## IN THE CLAIMS:

Please amend claims 1, 5, 7, 16, 17, 26, 34, 36, 45, 48 and 53. All pending claims are reproduced below.

1. (Currently Amended) A computer based system for switching between program contexts comprising:

a processor capable of having a first program thread and a second program thread in an execution pipeline having [[a]] thread selection hardware;

a first set of data storage devices capable of storing a first thread state of said processor;

a second set of data storage devices capable of storing a second thread state of said processor; and

a hardware thread scheduler for identifying which of said program threads said processor executes and configurable to allocate available processing time of the processor among at least the first and second program threads by causing thread-switching at a fixed time according to a predetermined fixed schedule, said schedule specifying that the first thread should be allocated processing time every first number of cycles and that the second thread should be allocated processing time every second number of cycles, wherein said first number of cycles is not equal to said second number of cycles.

2. (Previously Presented) The system of claim 17, wherein said first thread state is the thread state of the processor during the execution of the first program thread.

3. (Previously Presented) The system of claim 17, wherein said second thread state is the thread state of the processor during the execution of the second program thread.

4. (Previously Presented) The system of claim 17, wherein said processor switches between said first and second thread state by changing a state selection register.

5. (Currently Amended) The system of claim 17, wherein said hardware thread scheduler includes:

a thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread;

[[a]] an HRT scheduler for regularly scheduling ~~said~~ an HRT thread in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT thread in a predetermined time.

6. (Original) The system of claim 5, wherein said time quanta is at least one instruction cycle.

7. (Currently Amended) The system of claim 5, wherein said thread scheduler schedules a non-real-time (NRT) thread to replace a scheduled HRT thread if said scheduled HRT is idle.

8. (Original) The system of claim 5, wherein said thread scheduler schedules the execution of non-real-time (NRT) threads in quanta not allocated to HRT threads.

9. (Original) The system of claim 8, wherein said thread scheduler regularly schedules NRT threads to be executed.

10. (Original) The system of claim 5, further comprising:

a first storage device for storing program instructions, said processor fetching instructions from the first storage device within a first fetch period;

a second storage device for storing program instructions, said processor fetching instructions from the second storage device within a second fetch period;

wherein said first fetch period is substantially shorter than said second fetch period.

11. (Original) The system of claim 10, wherein said first storage device for storing program instructions comprises a static RAM.

12. (Original) The system of claim 10, wherein said second storage device for storing program instructions comprises a flash memory.

13. (Previously Presented) The system of claim 17, wherein said processor is capable of restoring said second thread state of said processor during execution of said first program thread.

14. (Previously Presented) The system of claim 17, wherein said processor is capable of storing said second thread state of said processor during execution of said first program thread.

15. (Previously Presented) The system of claim 17, wherein said first set of data storage devices comprises registers shared by a plurality of threads.

16. (Currently Amended) The system of claim 17, wherein the ~~fixed~~ execution schedule is one of a fixed strict schedule, a semi-flexible strict schedule, and a loose strict schedule.

17. (Currently Amended) A computer based system for switching between program contexts comprising:

a pipelined processor capable of having a first program thread and a second program thread in an execution pipeline having [[a]] thread selection hardware, the execution pipeline including a set of stages for executing instructions and configured to execute a single instruction at each different stage of the set of stages;

a first set of data storage devices capable of storing a first thread state of said pipelined processor;

a second set of data storage devices capable of storing a second thread state of said pipelined processor; and

a hardware thread scheduler for identifying which of said program threads said pipelined processor executes and configurable to allocate available processing time of the pipelined processor among at least the first and second program threads according to an execution schedule;

wherein said thread selection hardware in the pipelined processor switches from said first thread state to said second thread state between consecutive instruction cycles in response to the hardware thread scheduler identifying which of said program threads said pipelined processor executes.

18. (Original) The system of claim 5, wherein said time quanta is exactly one instruction cycle.

19. (Previously Presented) A computer based method for switching between program contexts in a multithreading pipelined processor having a hardware thread selector and an execution pipeline, the execution pipeline including a set of stages for executing instructions and configured to execute a single instruction at each different stage of the set of stages, the method comprising:

storing a first context of said pipelined processor in a first set of data storage devices, the first context corresponding to a first program thread;

storing a second context of said pipelined processor in a second set of data storage devices, the second context corresponding to a second program thread;

switching the pipelined processor from executing the first program thread to executing the second program thread between the end of an execution cycle and before the beginning of a next consecutive execution cycle by coupling the execution pipeline from the first set of data storage devices to the second set of storage devices via the hardware thread selector.

20. (Previously Presented) The method of claim 19, wherein the switching comprises changing a state selection register included in the hardware thread selector.

21. (Previously Presented) The method of claim 19, further comprising:

identifying which of the said program threads said processor executes according to an execution schedule.

22. (Previously Presented) The method of claim 21, further comprising:

allocating available processing time of the processor among at least the first and second threads according to the execution schedule.

23. (Original) The method of claim 22, wherein the allocating comprises dividing the available execution time into a plurality of quanta, each quanta corresponding to a number of instruction cycles for execution of a thread.

24. (Original) The method of claim 23, wherein at least one quantum corresponds to a thread that is scheduled to execute periodically after a fixed number of execution cycles.

25. (Original) The method of claim 21, wherein identifying further comprises identifying at least one hard-real-time (HRT) thread and at least one non-real-time (NRT) thread.

26. (Currently Amended) The method of claim 25, further comprising:

scheduling ~~the~~ an HRT thread in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT thread in a predetermined time.

27. (Original) The method of claim 25, further comprising:

scheduling an NRT thread for a quantum allocated for an HRT thread if said HRT thread is idle.

28. (Original) The method of claim 25, further comprising:

scheduling NRT threads in quanta not allocated for HRT threads.

29. (Previously Presented) The system of claim 1, wherein said thread selection hardware in the pipelined processor switches between said first and second thread state after the end of the execution of a first program instruction in the first thread and before the beginning of the execution of a second program instruction.

30. (Previously Presented) The system of claim 1, wherein said processor is an embedded pipelined processor.

31. (Previously Presented) The system of claim 1, wherein said first thread state is the thread state of the processor during the execution of the first program thread.

32. (Previously Presented) The system of claim 1, wherein said second thread state is the thread state of the processor during the execution of the second program thread.

33. (Previously Presented) The system of claim 1, wherein said processor switches between said first and second thread state by changing a state selection register.

34. (Currently Amended) The system of claim 1, wherein said hardware thread scheduler includes:

a thread identifier for identifying at least one hard-real-time (HRT) thread and at least one non-real-time thread;

a HRT scheduler for regularly scheduling ~~said~~ an HRT thread according to the predetermined fixed schedule in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT thread within a predetermined time.

35. (Previously Presented) The system of claim 34, wherein said time quanta is at least one instruction cycle.

36. (Currently Amended) The system of claim 34, wherein said hardware thread scheduler schedules a non-real-time (NRT) thread to replace a scheduled HRT thread if said scheduled HRT thread is idle.

37. (Previously Presented) The system of claim 34, wherein said hardware thread scheduler schedules the execution of non-real-time (NRT) threads in quanta not allocated to HRT threads.

38. (Previously Presented) The system of claim 37, wherein said hardware thread scheduler regularly schedules NRT threads to be executed.

39. (Previously Presented) The system of claim 34, further comprising:

a first storage device for storing program instructions, said processor fetching instructions from the first storage device within a first fetch period;

a second storage device for storing program instructions, said processor fetching instructions from the second storage device within a second fetch period;

wherein said first fetch period is substantially shorter than said second fetch period.

40. (Previously Presented) The system of claim 39, wherein said first storage device for storing program instructions comprises a static RAM.

41. (Previously Presented) The system of claim 39, wherein said second storage device for storing program instructions comprises a flash memory.

42. (Previously Presented) The system of claim 1, wherein said processor is capable of restoring said second thread state of said processor during execution of said first program thread.

43. (Previously Presented) The system of claim 1, wherein said processor is capable of storing said second thread state of said processor during execution of said first program thread.

44. (Previously Presented) The system of claim 1, wherein said first set of data storage devices comprises registers shared by a plurality of threads.

45. (Currently Amended) The system of claim 1, wherein the predetermined fixed schedule is one of a fixed strict schedule[[,]] or a semi-flexible strict schedule~~, and a loose strict schedule~~.

46. (Previously Presented) A computer based method for switching between program contexts in a multithreading pipelined processor having a hardware thread selector and an execution pipeline, the method comprising:

storing a first context of said processor in a first set of data storage devices comprising a first thread state corresponding to a first program thread;

storing a second context of said processor in a second set of data storage devices comprising a second thread state corresponding to a second program thread;

switching the processor from the first thread state to the second thread state by coupling the execution pipeline from the first set of data storage devices to the second set of

storage devices via the hardware thread selector at a fixed time according to a predetermined fixed execution schedule, said execution schedule specifying that the processor should switch to the first thread state every first number of cycles and that the processor should switch to the second thread state every second number of cycles, wherein said first number of cycles is not equal to said second number of cycles.

47. (Previously Presented) The method of claim 46, wherein the switching comprises changing a state selection register included in the hardware thread selector.

48. (Currently Amended) The method of claim 46, further comprising:

identifying which of the ~~said~~ program threads said processor executes according to a hard-real-time (HRT) execution schedule.

49. (Previously Presented) The method of claim 48, further comprising:

allocating available processing time of the processor among at least the first and second threads according to the predetermined fixed execution schedule.

50. (Previously Presented) The method of claim 49, wherein the allocating comprises dividing the available execution time into a plurality of quanta, each quanta corresponding to a number of instruction cycles for execution of a thread.

51. (Previously Presented) The method of claim 50, wherein at least one quantum corresponds to a thread that is scheduled to execute periodically after a fixed number of execution cycles.

52. (Previously Presented) The method of claim 48, wherein identifying further comprises identifying at least one hard-real-time (HRT) thread and at least one non-real-time (NRT) thread.

53. (Currently Amended) The method of claim 52, further comprising:

scheduling ~~the~~ a HRT thread in available time quanta such that said HRT thread is scheduled to ensure the execution of the HRT thread within a predetermined time.

54. (Previously Presented) The method of claim 52, further comprising:

scheduling an NRT thread for a quantum allocated for an HRT thread if said HRT thread is idle.

55. (Previously Presented) The method of claim 52, further comprising:

scheduling NRT threads in quanta not allocated for HRT threads.